Comparative Study of Machine Learning Classification Models on Loan Approval Prediction

Yufan Xie *

Department of Mathematics and Statistics, Kenyon College, Gambier, United States

* Corresponding Author Email: 2104098767leo@gmail.com

Abstract. As people's demands for loans with various purposes have increased drastically over the years, the process for banks to determine whether applications should be approved or not has become increasingly time-consuming and complicated. Therefore, models that can automatically provide banks with initial decisions for reference will significantly improve the efficiency of the process. Reviewing the results provided by the model may also offer an insight into potential biases in their decision-making procedures. The study will focus on determining better models based on evaluations and further manipulations of 6 machine learning methods: Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Random Forest, XGBoost, and Light-GBM. After comparisons among different classifiers established through different combinations of methods, such as oversampling and grid search, the study finally identifies two models that are most balanced on multiple metrics: the XGBoost model that undergoes oversampling and grid search, and a voting classifier, i.e., an ensemble of XGBoost, Random Forest, and Light-GBM models. In the comparison, it is found that, particularly when the original data set is imbalanced, the oversampling leads the models to make fewer conservative decisions on whether a loan application should be approved by decreasing the precision score while increasing the recall score.

Keywords: Loan Approval Prediction, Machine Learning, Binary Classification, Ensemble Learning, SMOTE.

1. Introduction

In financial system these days, loans are particularly meaningful for both banks and their customers. People of various backgrounds need loans for different purposes, and the interest and fees charged on the customers are also important to bank's profits. However, manual work to complete the task is tedious, and it is especially complicated when it comes to identifying a customer's probability of defaulting (i.e., failure to repay loans as agreed upon). Traditionally, banks primarily relied on one's credit score to assess the risk of defaulting [1]. In contrast, the rise in development of machine learning enables people to find latent patterns of customers' behaviors based on multiple features, which credit score may not be able to capture. Therefore, it is meaningful to test the accuracy of different machine learning models in this industry, which are useful for the banks and other financial organizations to evaluate their applicability.

There are many already published articles that propose a variety of procedures to construct ML models for the prediction of loan status. Besides the discrepancy on data sources, the differences among procedures of the model training also contribute significantly to the differences in results. These differences are especially helpful in revealing some insights into the constructions and applications of these models under different conditions. Though the existing library has already covered many approaches to establish better models in terms of multiple metrics, such as precision, recall, and accuracy scores, this study attempts to test new combinations of methods to provide new insight into how various practices interact with each other and thus affect the efficacy of the models at multiple aspects after considering different previous studies of related topics.

After going through the literature related to similar classification problem on loan approval prediction, the articles can be generally grouped based on the major techniques used in the research area. Surprisingly, the traditional methods are still of focus for a few authors to analyze and predict the loan defaults in financial area. For example, Sheikh et al. applied Logistic Regression after preprocessing with imputation and one-hot encoding [2], while Arutjothi & Senthamarai focused on

KNN combined with min-max normalization and outlier removal [3]. Therefore, they are still crucial baseline models considered in the comparative study.

In terms of other machine learning models' testing, people can observe that the tree-based models (e.g., Decision Tree, XGBoost, Random Forest, and LightGBM) are frequently used across different studies. Especially for the last 3 ensemble models, they often outperform most of the other models in classification. Tumuluru et al. compared Random Forest, SVM, KNN, and Logistic Regression, ultimately finding that Random Forest provided the highest accuracy [4]. In another extensive comparison, Serengil et al. evaluated a suite of models including XGBoost and LightGBM, concluding that LightGBM was the superior model [5]. Therefore, Decision Tree, Random Forest, XGBoost, and Light-GBM are selected in this study as the primary focus.

In terms of data preprocessing, most researchers decided to conduct feature engineering with practices determined by particular conditions of the datasets. This study also involves several related strategies, such as encoding and normalization (standard scaler) here. Nevertheless, several articles didn't include model tuning as part of the procedures, especially for more exhaustive hyperparameter tuning through grid search. Therefore, this study will conduct this practice for models in most conditions to evaluate its effect on the performances of final products.

In addition, most research on machine learning algorithms seldom covered the effect of imbalanced dataset on the predictions and how to address the problem, and they often focused more on its impact on the accuracy. For instance, an investigation on the deep learning applications on an extremely imbalanced dataset by Owusu E, Quainoo R, Mensah S, et al. primarily relies on the gap between the accuracy scores before and after oversampling (the score of DNN (Deep Neural Network) drops from 99% to 94%). However, few studies mentioned the reflection of imbalance in precision and accuracy scores [6]. Therefore, as the primary research purpose, the study would like to address the research gap. In this way, the author selected the common metrics, including accuracy, precision, recall score, and F1-score (i.e., a comprehension of both precision and recall). They can generally generalize the model performances at multiple aspects to avoid the bias in answering the research question.

At last, this study was intended to test the performance of more complicated models either, such as deep learning algorithms (e.g., LSTM and transformer) particularly, under difference conditions. However, due to the limitations in available devices, the study switches to some other related research on similar topics to compare their results with the performances of the less complicated ML algorithms. After reading through the related literature, the most comprehensive study of relevant topic found is conducted by Hussein Sayed E, Alabrah A, Hussein Rahouma K, et al. to demonstrate the impacts of SMOTE and ensemble learning techniques on deep learning algorithms' predictions at multiple aspects [7]. More details of the comparison will be discussed later in discussion section.

2. Methodology

This research was done on a local Jupyter Notebook Environment in Python. The models predict whether a customer is eligible to receive loans. The following sections discuss basic information about the dataset used and each step of the procedure.

2.1. Dataset

The data source used in the study is the dataset, "Loan Approval Classification Dataset," available on Kaggle [8]. It contains more than 40,000 cases of decisions on loan status. Table 1 below gives the basic conditions of the data set. Here, the credit score is removed from the analysis, since the test on the effect of its presence indicates that it fails to have a significant effect on the model performances (the credit score is commonly computed based on several other features in the data set).

Table 1. Overview of features in the dataset

Variable Name	Description	Data Type
person_age	Age of the person	Float
person_gender	Gender of the person	Categorical
person_education	Highest education level	Categorical
person_income	Annual income	Float
person_emp_exp	Years of employment experience	Integer
person_home_ownership	Home ownership status (e.g., rent, own, mortgage)	Categorical
loan_amnt	Loan amount requested	Float
loan_intent	Purpose of the loan	Categorical
loan_int_rate	Loan interest rate	Float
loan_percent_income	Loan amount as a percentage of annual income	Float
cb_person_cred_hist_length	Length of credit history in years	Float
credit_score	The credit score of the person	Integer
previous_loan_defaults_on_file	Indicator of previous loan defaults	Categorical
loan_status	(target variable) Loan approval status: 1 = approved; 0 = rejected	Integer

2.2. EDA

To explore the general patterns and trends of each feature's distribution, the author conducted Explanatory Data Analysis (EDA) through basic descriptive statistics tables of numeric and categorical features (table 2 and table 3).

Table 2. Descriptive Statistics of Numeric Feature

	person_age	person_income	person_emp_exp	loan_amnt	loan_int_rate	loan_percent_income	cb_person_cred_hist_length	credit_score
count	45000.00	45000.00	45000.00	45000.00	45000.00	45000.00	45000.00	45000.00
mean	27.76	8.031905e+04	5.41	9583.16	11.01	0.14	5.87	632.61
std	6.05	8.042250e+04	6.06	6314.89	2.98	0.087	3.88	50.44
min	20.00	8.000000e+03	0.00	500.00	5.42	0.00	2.00	390.00
25%	24.00	4.720400e+04	1.00	5000.00	8.59	0.070000	3.00	601.00
50%	26.00	6.704800e+04	4.00	8000.00	11.01	0.120000	4.00	640.00
75%	30.00	9.578925e+04	8.00	12237.25	12.99	0.190000	8.00	670.00
max	144.00	7.200766e+06	125.00	35000.00	20.00	0.66	30.00	850.00

Table 3. Descriptive statistics of categorical features

	person_gender	person_education	person_home_ownership	loan_intent	previous_loan_defaults_on_file
count	45000	45000	45000	45000	45000
unique	2	5	3	6	2
top	male	Bachelor	RENT	EDUCATION	N
freq	24841	13399	23443	9153	36858

Figure 1 indicates right skewness in the distributions of person_income (personal income) and loan_amnt (loan amount) through the obvious difference between the mean and median (50%). In addition, the loan amounts are overall much larger than other numeric features in number. Therefore,

the large figures can potentially allow the feature to gain more unexpected weight while being input in the model fitting. According to Figure 1, since the skewness is not overwhelmingly large, as indicated by the moderate distance between means and medians, it is appropriate to ignore the conditions temporarily. However, it is necessary to normalize all of these distributions to get them on similar scales, which can improve the performance of the models in real-world applications.

2.3. Data Preprocessing

The procedure of data processing in the study is then designed, given the information provided by EDA and descriptions of the dataset. To optimize the performance of the models, the study includes several practices of feature engineering to process the data.

2.3.1. Standard Scaler

To account for unexpected weight on certain variables brought by the large figures, the author selected a standard scaler to normalize the distribution of each numeric feature (i.e., float or integer) except loan_status, which will be later encoded. The standard scaler functions by scaling each *x* value in the columns in z-score with the following formula:

$$z = (x - \mu)/\sigma \tag{1}$$

After the transformation, all of the data are in a z-distribution with a mean of 0 and a standard deviation of 1. As all of the features are measured on the same scale, the unexpected weight on features originally measured in large figures is prevented.

2.3.2. OneHotEncoder

Since the machine learning algorithm cannot understand the differences between pure texts, different categories of each nominal variable (i.e., categorical features that don't have the inherent order) need to be encoded into numbers in a specific scale for the machine to identify. Specifically, the encoder recognizes all of the unique categories of a feature and adds a new binary feature in memory for a unique category. For the row of a particular category, its binary feature will be set to 1 for this row. Therefore, the algorithm can identify which category each customer falls into. In the dataset of loan applications, person_gender, loan_intent, and previous_loan_defaults_on_file are all nominal features, which need to be encoded through the one-hot encoder built in the scikit-learn package.

2.3.3. Ordinal Encoder

Besides the nominal features, there are ordinal features in the dataset, which have inherent orders among categories. Hence, they need to be encoded through an ordinal encoder rather than a one-hot encoder to emphasize the inherent order. Instead of coding each category through binary features, the ordinal encoder assigns a unique number to each category, and the indices will follow the order of variables assigned by the programmer in a list.

2.3.4. SMOTE (Oversampling)

According to the bar plot of loan_status, less than 30% of the data falls in the "approval" (1) category. When the minority class is limited in amount, models may fail to provide enough information for training of the algorithm. For instance, the historical data of bankrupt companies are frequently impacted by imbalanced data, where the main data source is the non-bankrupt companies, leading to erroneous predictions [9]. Here, the technique of SMOTE (oversampling), as mentioned in previous literature, is highly effective in balancing the sizes of two categories by creating synthetic examples based on existing data [10]. It randomly chooses one of its neighbors and creates new data along the distance between this neighbor and itself. Therefore, more information provided in the minority class may probably be less biased to the majority class. The effects of the method will be further discussed in later sections.

2.4. Models

To evaluate models from multiple perspectives, several models across different categories are selected for the prediction on loan status, as shown in Table 1. Logistic regression is one of the typical representatives of linear models; decision tree is a frequently used tree-based model. The K-NN is an instance-based learning method, as it determines the response based on the similarity between already existing instances and a new instance. Beyond the individual algorithms, ensemble models, such as XGBoost and Random Forest, are also included in the analysis procedure. Here are the descriptions of each model and their potential contributions in the study.

2.4.1. Logistic Regression

Logistic regression is a classic statistical method used for binary classification problems. It can predict the probability from 0 to 1 for an instance to fall in a certain category. Here is the function that takes the input and outputs the probability.

$$\sigma(z) = 1 / (1 + e^{-z}) \tag{2}$$

To answer the classification problems, normally, the threshold of its decision is at 0.5 by default, and the analysis of the study also takes the threshold.

2.4.2. K-NN

K-Nearest Neighbors (K-NN) is also a model that can be used for classification problems. Instead of directly building the algorithms, it retains the original data for training and compares the new instances with the original cases to make a decision. It determines the nearest neighbors of a new instance through the distance formula below.

$$d(p, q) = \sqrt{\sum_{i=1}^{n}(pi - qi)^2} \tag{3}$$

Therefore, it assigns the label of the most neighbors selected based on the distance to the new instance.

2.4.3. Decision Tree

Decision tree is one of the most important tree-based models in machine learning. It learns to classify a case by determining the thresholds based on previous data and developing leaf nodes to mimic human decision-making. To make a prediction for a new data point, you start at the root node and follow the path down the tree based on the evaluations with preset thresholds.

2.4.4. Support Vector Machine

SVM, another commonly used machine learning model, decides on classification by the hyperplane that it develops to separate the data into two categories. In this process, it intends to maximize the margin, the overall distance between the hyperplane and most instances, to be more certain about its results. Here is the linear equation of the hyperplane to divide the space:

$$w \cdot x - b = 0 \tag{4}$$

However, SVM is very flexible in using lines or planes that are not straight to categorize the data in a nonlinear way, when the operation is necessary. The study will only consider the line model to classify the instances in the test set for simplicity.

2.4.5. Random Forest

Random Forest is an ensemble learning algorithm that combines multiple decision trees to make better decisions. Each tree in the forest is trained on a slightly different subset of the original training data. Then, the model as a whole determines its answer by gathering the votes from all of the individual decision trees.

2.4.6. XGBoost

Similarly, XGBoost is another ensemble model that builds multiple trees for decision-making. Nevertheless, instead of building independent trees, XGBoost trains a single tree at each step and builds the new trees considering the error made by the previous tree model, which is one of the important implementations of gradient boosting. The formula below determines whether the model training should build another split of trees.

$$\mathrm{Obj}^{(t)} \approx \sum_{i=1}^{n}[g_i f_t(x_i) + 1/2 \cdot h_i f_t^2(x_i)] + \Omega(f_t) \quad (5)$$

Based on the errors and complexity of the current model computed through the formula, the algorithm can balance the goals to minimize the error in the prediction of the training set and to avoid overfitting.

2.4.7. Light-GBM

Light-GBM, the abbreviation of "Light Gradient-Boosting Machine," works following a similar procedure with XGBoost model: they both seek to improve the model by building new trees based on the performances of previous trees. However, as XGBoost establishes the trees in a balanced way regardless of whether a tree has a particularly large gradient, LightGBM focuses more on the side that demands more corrections (i.e., the side with large gradient). Therefore, it is more efficient and faster than XGBoost, since it doesn't attempt to capture all of the possibilities but constructs an unbalanced collection of trees. Therefore, it computes the gain with weights to allocate the attention differently to two sides through the formula below.

$$Gain_{GOSS} = \frac{1}{2}\left[\frac{\left(G_L^A + w \cdot G_L^B\right)^2}{H_L^A + w \cdot H_L^B + \lambda} + \frac{\left(G_R^A + w \cdot G_R^B\right)^2}{H_R^A + w \cdot H_R^B + \lambda} - \frac{\left(G_P^A + w \cdot G_P^B\right)^2}{H_P^A + w \cdot H_P^B + \lambda}\right] - \text{у} \quad (6)$$

However, it is worth noting that Light-GBM has a higher risk of running into overfitting, since it considers a smaller sample of trees. Fortunately, as the sample size of the data set for this study is sufficiently large (i.e., more than 40,000 rows), the risk is less concerning because of the diversity of the dataset.

2.5. Model Tuning

Individual models are usually not sufficient to reach the expected performance in prediction for the test set. The practice of testing different combinations of hyperparameters for each model through strategies of model tuning is often effective in improving the performances. Here, the grid search (GridSearchCV) is a common practice to complete the task conveniently with a few lines of code. Here is an overview of the hyperparameters considered in the model tuning procedure in Table 2.

Table 4. The hyperparameter grid of each selected model

Model	Hyperparameter	Values Tested
Random Forest	n_estimators	[100, 200]
	max_depth	[5, 10, 15]
XGBoost	n_estimators	[100, 200, 300]
	max_depth	[3, 5, 7]
	learning_rate	[0.01, 0.1]
	subsample	[0.7, 1.0]
Light-GBM	n_estimators	[100, 200, 300, 400]
	max_depth	[3, 5, 7, 9]
	learning_rate	[0.01, 0.05, 0.1]
	subsample	[0.7, 1.0]

3. Results

To evaluate the performance of classification models, the study determines the confusion matrix (precision score and recall score), F1 score, and accuracy as the metrics. The displays of metrics of models that follow are all in terms of the performances on positive predictions (to approve the loans).

As the comparison between models trained by an imbalanced and balanced dataset is one of the key discussions of the research, the oversampling would not be applied to the dataset for the first few steps.

The individual models are first tested to determine the three best-fit models for further operations to improve their results. Here is an overview of the performances of all of the individual models in table.

Table 5. Overview of the performances of individual models

Model	Precision	Recall	F1-score	Accuracy
Logistic Regression	0.76	0.74	0.75	0.89
SVM	0.76	0.73	0.75	0.89
Decision Tree	0.84	0.73	0.78	0.91
KNN	0.83	0.67	0.74	0.9
Random Forest	0.89	0.76	0.82	0.93
XGBoost	0.88	0.75	0.81	0.92
Light-GBM	0.88	0.78	0.83	0.93

According to the results above in Table 3, all of the models have about the same accuracy score. That indicates a uniformly outstanding performance in percent of correct predictions out of the total number of instances. The accuracy metric is deceptive, as less than 30% of the data points are marked as approved on the loan applications. Therefore, a benchmark model that marks each customer as ineligible will get at least 70% on accuracy. In that case, the precision score and recall score can provide more information on the models' performances.

Since either precision score or recall score cannot stand for a model's overall performance, the F1-score (harmonic mean of recall and precision scores) is a more helpful metric to generalize the proficiency. Among these 7 models selected, the random forest, XGBoost, and Light-GBM models are clearly the best models with higher F1-scores. Therefore, overall, they are more effective both in terms of the correct rate within the positive predictions and the proportion of correct positive predictions among the actually positive results. In conclusion, through the comparison, the study decides to proceed with the random forest, XGBoost, and Light-GBM models for model tuning, which would hopefully improve their performances.

Here are the performances of the best models after the grid searches on every possible combination of hyperparameters for each individual model (the hyperparameter grid for tuning is shown in the methodology section). As mentioned before, since the overall performance demonstrated by both precision and recall scores is of more importance in the study, the F-1 score is considered as the only metric for the grid search procedure to select the best model.

Table 6. Overview of the best models selected by grid searches (before oversampling)

Model	Precision	Recall	F1-score	Accuracy
Random Forest	0.90	0.75	0.82	0.92
XGBoost	0.89	0.80	0.84	0.93
Light-GBM	0.88	0.80	0.84	0.93

According to the results above in Table 4, XGBoost now has the best performance among 3 models, as it has balanced precision and recall scores. Light-GBM is also a very strong competitor. The balanced performance in two separate metrics contributes to their higher F1-scores, as shown in the table. In contrast, the performances of random forest models clearly demonstrate the potential impact of an imbalanced dataset: as there are much more instances that

are marked as ineligible to receive the loans compared with the ones approved, the models constructed are prone to be very conservative on approving the customers' loan applications. The tendency is also reflected in their high precision score and poor performance on the recall score. That is, they tend to consider avoiding loan defaults as the higher priority and sometimes reject applications that should be considered qualified in the original procedure.

The output of grid searches after oversampling applied to the dataset provides firm proof of the claim. Here is an overview of the performances of the best models selected by grid searches after oversampling is applied in Table 5.

Table 7. Overview of the best models selected by grid searches (after oversampling)

Model	Precision	Recall	F1-score	Accuracy
Random Forest	0.75	0.87	0.81	0.91
XGBoost	0.86	0.81	0.83	0.93
Light-GBM	0.86	0.81	0.84	0.93

After the dataset is perfectly balanced, it is obvious that the random forest model entirely changes its tendency to predict the loan status. With more information related to the positive side, they now consider enhancing the recall score as the higher priority by trying to cover all of the eligible customers in predictions. In consequence, the precision scores of both models are much lower than their recall scores under the situation. Nonetheless, XGBoost and Light-GBM models, as compared with the other two models, still retain stable performance on both metrics (0.86 on the precision score and 0.81 on the recall score).

4. Discussion

According to the results above, the comparative study of different models on their performances in dealing with either balanced or imbalanced dataset provides important insight into the superiority of certain models in dealing with particular conditions. Here, when testing the individual models without considering oversampling and grid search, XGBoost, random forest, and Light-GBM didn't demonstrate stark discrepancies on their performance on 4 metrics considered (i.e., precision, recall, F1-score, and accuracy). However, three algorithms' bests model selected based on F1-score in grid searches vary tremendously on the same metrics. XGBoost and Light-GBM models consistently show balanced performances on two criteria: while they are excellent on the correct rate of identifying the customers that should be approved on loans, they still maintain an expected performance on capturing as many eligible customers as possible. Nevertheless, random forest fell short on the expected balance in two metrics. In particular, three individual models that outperform others are all tree-based models, which is a convincing proof of tree-based models' prestige on loan predictions (classification). However, among the tree-based models, they also belong to two different families, and XGBoost and Light-GBM algorithms build the decision trees based on gradient boosting rather than bagging. Therefore, the conclusion that the gradient-boosting algorithms can remain more consistent and balanced performances in precision and recall scores is plausible.

In terms of the effectiveness of oversampling technique to address the imbalance in the data, as generalized in result section, after SMOTE is applied to the training set, the tendency of the performances of random forest on precision and recall metrics completely switch: before oversampling, the best model selected by grid search has an extremely high precision of 0.9, which is compensated by unexpectedly low recall score of 0.75. After SMOTE applied, its precision immediately drops to 0.75, while its recall score rises to 0.87. As the model tends to capture more positive predictions (i.e., the eligible customers) after oversampling, it is reasonable to conclude that more data in the minority class helps the models to be more certain about whether an instance should be approved. They are inclined to be less conservative on its decisions then in most cases.

Hussein Sayed E, Alabrah A, Hussein Rahouma K, et al.'s study on the comparative study of machine learning and deep learning models in the conditions with and without SMOTE applied further strengthens the conclusion [7]. According to his report on models' performances on different metrics, the deep learning algorithms similarly present a wide gap in overall performance before and after oversampling technique is applied to deal with the imbalanced dataset for loan default classification. The best DL models in the study, ResNet and DenseNet, achieve around 0.8 on precision score and 0.7 on the recall score before oversampling, which is obviously unbalanced as expected by the results of random forest model in Table 4. With the balanced technique of SMOTE, their performances on both metrics reach the balance on 0.8, which come with improved F1-score. However, the best model, without considering ensemble learning method, is still random forest, which outperforms both gradient boosting model and more complex deep learning algorithms. Therefore, despite shared finding of efficacy of oversampling technique on leading the models performances to be more balanced, the best-fit models of a study still depend on the dataset used in addition to the topic and category of the problem. Similar counterpoint against the superiority of tree-based is also proposed by Fati by presenting the logistic regression, one of the most simplest classification algorithm, to be the best model among all [11].

After all, even on the similar topic of loan prediction, different factors, such as the data source, analysts' operations, model choices, and imbalance, are all influential in the final results of the models. Here, a single valid answer on which is the best model cannot be definite, and it depends on people's demands: whether they prefer a model more conservative on approving the loans to avoid loan defaults or a model of more balanced on approval or rejections.

5. Conclusion

In this study, 6 individual models (logistic regression, KNN, SVM, decision tree, random forest, and XGBoost) are used to predict whether a customer is eligible to receive loans in the banking industry. Among them, three of the tree-based models have the best performances and were selected for the practices of model tuning and comparisons on the performances before and after oversampling being applied. Ultimately, the XGBoost model selected by grid search based on F1-score outperforms others and retains a balance between precision score and recall score consistently. Hence, it is the best model among all of the remaining options from this perspective. It is also worth discussing that the study provides convincing support for the important impact of oversampling on the models' performances. With more information provided in the original minority class, the models tend to improve on the recall score by attempting to identify all of the eligible customers for more potential profits. However, that doesn't mean the models will definitely be more suitable for real-world applications. People still need to select the models based on their specific demands.

However, this study still falls short on several aspects. As shown by previous literature, there are still a lot of models with different emphases that the study fails to cover. In addition, because of time constraints and the limited number of features, the study doesn't go through all of the features to conduct careful selections based on comparisons of models in multiple conditions. Besides, since undersampling will significantly decrease the sample size of the dataset, it is not taken for testing in the study. In future studies based on the larger datasets, the comparison between the influences of undersampling and oversampling is also a meaningful topic to be investigated. At last, as mentioned before, because of the time and devices constraints, the study didn't directly apply more complex models, such as deep learning algorithms, to the dataset. The straightforward comparison of models on the same dataset with all of other analysis operations controlled will definitely promote a more convincing conclusion on the differences made by the complexity of models and other available techniques. In conclusion, there is still a lot of room left for later studies to further develop.

References

[1] Mamun M A, Farjana A, Mamun M. Predicting bank loan eligibility using machine learning models and comparison analysis. 7th North American International Conference on Industrial Engineering and Operations Management, 2022: 1423–1432.

[2] Sheikh M A, Goel A K, Kumar T. An approach for prediction of loan approval using machine learning algorithm. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020: 490–494.

[3] Arutjothi G, Senthamarai C. Prediction of loan status in Commercial Bank using machine learning classifier. 2017 International Conference on Intelligent Sustainable Systems (ICISS), 2017: 416–419.

[4] Tumuluru P, Burra L R, Loukya M, et al. Comparative analysis of Customer Loan Approval Prediction using machine learning algorithms. 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), 2022: 349–353.

[5] Serengil S I, Imece S, Tosun U G, et al. A comparative study of machine learning approaches for non performing loan prediction. 2021 6th International Conference on Computer Science and Engineering (UBMK), 2021: 326–331.

[6] Owusu E, Quainoo R, Mensah S, et al. A deep learning approach for loan default prediction using Imbalanced Dataset. International Journal of Intelligent Information Technologies, 2023, 19(1): 1–16.

[7] Hussein Sayed E, Alabrah A, Hussein Rahouma K, et al. Machine learning and deep learning for loan prediction in banking: Exploring Ensemble Methods and data balancing. IEEE Access, 2024, 12: 193997–194019.

[8] Ta-Wei L. Loan Approval Classification Dataset. Kaggle, 2021.

[9] El Madou K, Marso S, El Kharrim M, et al. Evolutions in machine learning technology for Financial Distress Prediction: A comprehensive review and comparative analysis. Expert Systems, 2023, 41(2).

[10] Orji U E, Ugwuishiwu C H, Nguemaleu J C, et al. Machine learning models for predicting bank loan eligibility. 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), 2022: 1–5.

[11] Fati S M. Machine learning-based prediction model for loan status approval. J. Hunan Univ. Nat. Sci., 2021, 48(10).